

Contents

| | |
|--|----------|
| 1 Routine/Function Prologues | 2 |
| 1.0.1 time_interp_geos.F90 (Source File: time_interp_geos.F90) | 2 |
| 1.1 Core Functions of time_interp_geos | 2 |

1 Routine/Function Prologues

1.0.1 time_interp_geos.F90 (Source File: time_interp_geos.F90)

Opens, reads, and interpolates GEOS forcing.

TIME1 = most recent past data

TIME2 = nearest future data

The strategy for missing data is to go backwards up to 10 days to get forcing at the same time of day.

1.1 Core Functions of time_interp_geos

zterp Performs zenith angle-based temporal interpolation

REVISION HISTORY:

```

1 Oct 1999: Jared Entin; Initial code
25 Oct 1999: Jared Entin; Significant F90 Revision
11 Apr 2000: Brian Cosgrove; Fixed name construction error
              in Subroutine ETA6HFILE
27 Apr 2000: Brian Cosgrove; Added correction for use of old shortwave
              data with opposite sign convention from recent shortwave data.
              Added capability to use time averaged shortwave & longwave data
              Altered times which are passed into ZTERP--used to be GMT1
              and GMT2, now they are LDAS%ETATIME1 and LDAS%ETATIME2
30 Nov 2000: Jon Radakovich; Initial code based on geteta.f
17 Apr 2001: Jon Gottschalck; A few changes to allow model init.
13 Aug 2001: Urszula Jambor; Introduced missing data replacement.
5 Nov 2001: Urszula Jambor; Reset tiny negative SW values to zero.

```

INTERFACE:

```
subroutine time_interp_geos()
```

USES:

```

use lisdrv_module, only : lis, grid
use baseforcing_module, only : glbdata1, glbdata2
use time_manager
use grid_spmdMod
use time_module, only : time2date
use spmdMod
use geosdomain_module, only : geosdrv

```

CONTENTS:

```

if(masterproc) then
  if (get_nstep() .eq. 0) then
    lis%f%force = lis%f%nmif
  else

```

```

        lis%f%nforce = lis%f%nf
    endif
endif
#endif(defined SPMD)
call MPI_BCAST(geosdrv%geostime1,1,MPI_REAL8,0, &
    MPI_COMM_WORLD,ier)
call MPI_BCAST(geosdrv%geostime2,1,MPI_REAL8,0, &
    MPI_COMM_WORLD,ier)
call MPI_BCAST(lis%t%time,1,MPI_REAL8,0, &
    MPI_COMM_WORLD,ier)
call MPI_BCAST(lis%t%gmt,1,MPI_REAL,0, &
    MPI_COMM_WORLD,ier)
call MPI_BCAST(lis%f%nforce,1,MPI_INTEGER,0, &
    MPI_COMM_WORLD,ier)
call MPI_BCAST(lis%f%shortflag,1,MPI_INTEGER,0, &
    MPI_COMM_WORLD,ier)
call MPI_BCAST(lis%f%longflag,1,MPI_INTEGER,0, &
    MPI_COMM_WORLD,ier)
#endif
btme=geosdrv%geostime1
call time2date(btme,bdoy,gmt1,byr,bmo,bda,bhr,bmn)
btme=geosdrv%geostime2
call time2date(btme,bdoy,gmt2,byr,bmo,bda,bhr,bmn)
!-----
! Interpolate Data in Time
!-----
wt1=(geosdrv%geostime2-lis%t%time)/
    (geosdrv%geostime2-geosdrv%geostime1)
wt2=1.0-wt1

do f=1,lis%f%nforce
    if(f.eq.3) then
        if (lis%f%shortflag.eq.2) then
!-----
! Got Time Averaged SW
!-----
        do c=1,gdi(iam)
            zdoy=lis%t%doy
            call zterp(0,grid(c)%lat,grid(c)%lon, &
                gmt1,gmt2,lis%t%gmt,zdoy, &
                zw1,zw2,czb,cze,czm,lis)
            grid(c)%forcing(f)=glbdata2(f,c)*zw1

            if ((grid(c)%forcing(f).ne.lis%d%udef).and. &
                (grid(c)%forcing(f).lt.0) ) then
                if (grid(c)%forcing(f) > -0.00001) then
                    grid(c)%forcing(f) = 0.0
                else

```

```
      print*, 'ERR: time_interp_geos -- Stopping because ', &
              'forcing not udef but lt0,'
      print*, 'ERR: time_interp_geos -- ', &
              'f,c,grid(c)%forcing(f),glbdata2(f,c)', &
              f,c,grid(c)%forcing(f),glbdata2(f,c), &
              ',(,iam,)'
      call endrun
    end if
  endif

  if (grid(c)%forcing(f).gt.1367) then
    grid(c)%forcing(f)=glbdata2(f,c)
  endif
enddo
endif

else if(f.eq.8.or.f.eq.9) then
!-----
! precip variable Block Interpolation
!-----
  do c=1,gdi(iam)
    grid(c)%forcing(f)=glbdata2(f,c)
  enddo

  else if (f.eq.4) then
    if (lis%f%longflag.eq.1) then
!-----
!      Got Instantaneous LW
!-----
      do c=1,gdi(iam)
        grid(c)%forcing(f)=glbdata1(f,c)*wt1+ &
          glbdata2(f,c)*wt2
      enddo
    endif

    if (lis%f%longflag.eq.2) then
!-----
!      Got Time Averaged LW
!-----
      do c=1,gdi(iam)
        grid(c)%forcing(f)=glbdata2(f,c)

      enddo
    endif

  else
!-----
!      Linearly interpolate everything else
!-----
```

```
!-----
      do c=1,gdi(iam)
        grid(c)%forcing(f)=glbdata1(f,c)*wt1+ &
          glbdata2(f,c)*wt2
      enddo
      endif
    enddo
84  format('now',i4,4i3,2x,'pvt ',a22,' nxt ',a22)
      return
```